

数 据 库

课程设计说明书

题 目： 学生学籍信息管理系统

院 系： _____

专业班级： _____

学 号： _____

学生姓名： _____

指导教师： _____

2010 年 5 月 31 日

学 号		学 生 姓 名		专 业（班级）	
设计题目	学生学籍信息管理系统				
设计技术参数	数据库： <u>MYSQL</u> 开发语言： <u>JAVA</u> (1) 主要的数据表： 学生信息数据表，专业班级数据表，系统用户数据表等。 (2) 主要功能模块 系统管理：管理员帐号添加、修改、删除、密码修改等。 专业班级管理：实现专业班级的添加、删除、修改、查询等功能。 学生信息管理：实现学生信息的添加、删除、修改、查询等功能。 学生检索：要求提供多种检索方式。				
设计要求	(1) 三到四人为一个小组，小组成员既要有团队协作精神，又要分工明确。每个学生都必须充分了解整个设计的全过程。 (3) 从开始的系统需求分析到最后的系统测试，都要有详细的计划，设计文档应按照软件工程的要求书写。 (3) 系统中的数据表设计应合理、高效，尽量减少数据冗余。 (4) 系统安全性高。				
工作量	课程设计说明书字数要求 3000 以上，不包括图表。				
工作计划	第 2 周：需求分析,总体设计 第 3～9 周：详细设计,上机调试 第 10～11 周：编写课程设计说明书及验收				
参考资料	[1] 王珊,萨师煊. 数据库系统概论[M]. 北京:高等教育出版社,2007. [2]Java 语言程序设计 北京：清华大学出版社,2005. [3]SQL Server 2000实验指导（第 2 版）北京：清华大学出版社,2008.				
指导教师签字				教研室主任签字	

学生姓名：_____ 学号：_____ 专业班级：_____

课程设计题目：_____ 学生学籍信息管理系统

指导教师评语：

成绩：_____

指导教师：_____

年 月 日

目 录

1、问题描述	错误!未定义书签。.....
主要的数据库表:	错误!未定义书签。.....
主要功能模块	错误!未定义书签。.....
2、可行性研究.....	错误!未定义书签。.....
3、需求分析	错误!未定义书签。.....
4、总体设计	错误!未定义书签。.....
主要表的设计:	错误!未定义书签。.....
学生学籍信息管理系统流程图.....	错误!未定义书签。
数据库关系图	错误!未定义书签。.....
5、详细设计	错误!未定义书签。.....
主菜单设计.....	错误!未定义书签。.....
函数设计及主要功能描述	错误!未定义书签。
6、编码和单元测试	错误!未定义书签。.....
7、综合测试	错误!未定义书签。.....
各函数测试结果	错误!未定义书签。.....
测试	错误!未定义书签。.....
8、软件维护	错误!未定义书签。.....
9、总结	错误!未定义书签。.....
10、参考文献.....	错误!未定义书签。.....

1、问题描述

主要的数据库表：

学生信息数据表，专业班级数据表，系统用户数据表等。

主要功能模块

系统管理：管理员帐号添加、修改、删除、密码修改等。

专业班级管理：实现专业班级的添加、删除、修改、查询等功能。

学生信息管理：实现学生信息的添加、删除、修改、查询等功能。

学生检索：要求提供多种检索方式。

2、可行性研究

对于所要求的功能模块，可以通过 **JAVA**程序连接到数据库。通过相对应的操作要求，可以通过用户或管理员的方式登录，然后通过提示完成上述一系列的操作，包括：账号的添加、修改，学生信息的添加、修改、删除，班级信息的添加、修改、删除等等操作。

3、需求分析

用户登录：操作员可以分为两种，一种是管理员，一种是普通用户。两者的权限是不一样的，所以在通过不同身份进入到系统里面会出现不同的操作界面。其中管理员多了添加、修改、删除操作，还有就是用户管理操作；而一般的用户只有查看和修改自己密码的权限。

查看当前信息：主要是通过对数据库的访问，查看当前数据库中的信息是否完整，然后才能进行其它操作。这一功能是普通用户和管理员都有的权限，所以两个方法可以合二为一，而不用去分别写了。查看的信息包括：学生信息和班级信息。管理员还包括一个用户信息查询。当然也可以对个别学生通过不同的查找方式来查看对应学生的信息和班级信息。

修改密码：管理员可能通过不同的方法对自己的密码和其它所有用户的密码进行修改。而普通用户只能通过此功能修改自己的密码。

用户管理：管理员特权。其中包括：创建用户、删除用户、修改用户密码等。通过对已经用户的查看，如果有必要创建一个新用户的话就可以通过此方法来创建。当然用户名相同的不能再次创建。如果有用户不再使用此系统，为保证安全

性可以尽快将其删除。而修改密码是和上一个方法联系在一起的。

插入操作：管理员特权！在系统之初，数据库里面可能包含太少的学生信息和班级信息。而管理员则可以此方法向数据库中添加数据。其中包括添加学生信息和班级信息。

删除操作：管理员特权！如果系统用的时间比较长的话，那么学生信息和班级信息不可能一直不变。如果出现了学生离校或者班级编排方式不同，则有可能出现多余的信息，此时可以通过此方法将多余的信息删除掉。其中包括删除学生信息和班级信息。

修改操作：管理员特权！在管理员添加信息的时候，出错是难以避免的，而有时候一个学生的信息只是错了个班级。如果将其删除然后再重新添加费时又费力。相反可以通过此方法对一条记录中的个别错误进行修改。这样可以节约很多添加和删除操作。其中包括修改学生信息和班级信息。

主函数：主要用于整体框架的构建。并通过输入说明将方法分解开来，因此，用户可以完成准确的操作。

4、总体设计

主要表的设计：

此系统包括三个表：学生学籍信息表（图 4-1）、班级信息表（图 4-2）、用户表（图 4-3）

设计表“student”，位置是“Student”中、“(local)”上				
	列名	数据类型	长度	允许空
	sno	char	10	
	sname	char	8	✓
	ssex	char	2	✓
	sclass	smallint	2	✓
	sage	int	4	✓
	sdept	smallint	2	✓

图 4-1

设计表“class”，位置是“Student”中、“(local)”上				
	列名	数据类型	长度	允许空
	cno	char	10	
	cname	char	10	<input type="checkbox"/>
	cdept	char	20	✓
	cmonitor	char	10	✓

图 4-2

设计表“users”，位置是“Student”中、“(local)”上

	列名	数据类型	长度	允许空
	username	char	20	
	password	char	15	✓

图 4-3

学生学籍信息管理系统流程图（图 4-4）

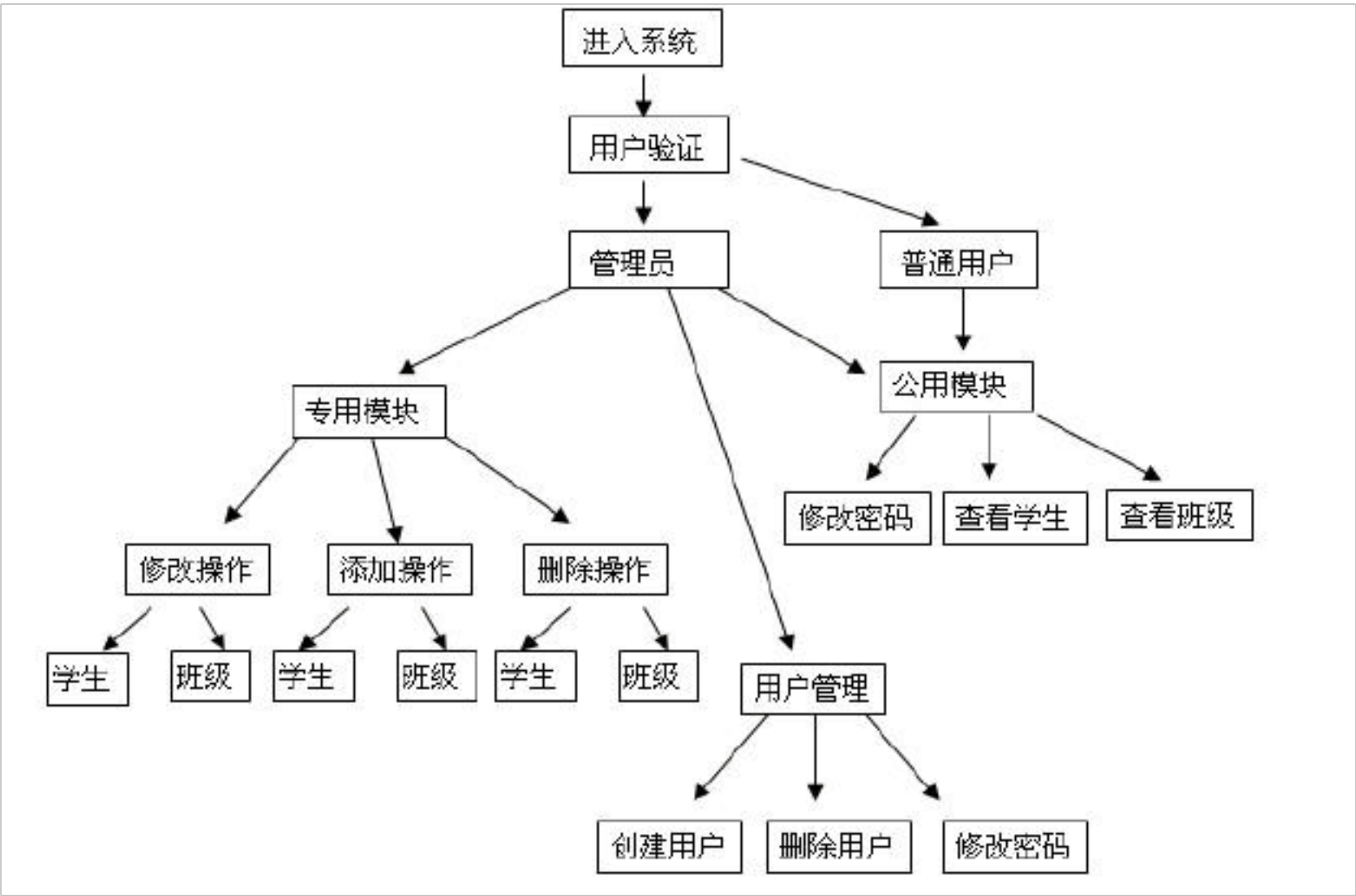


图 4-4

数据库关系图(图 4-5)

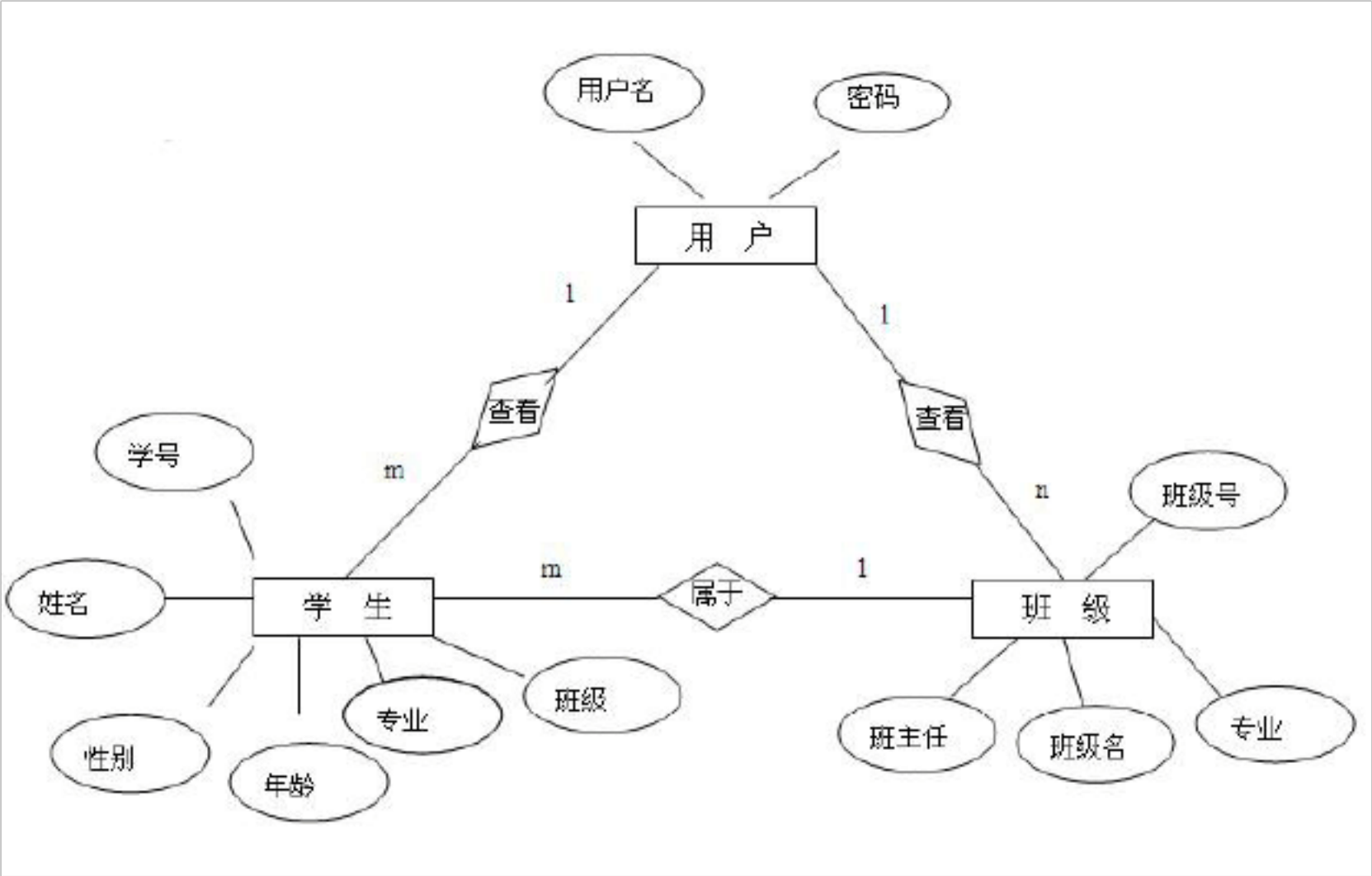


图 4-5

5、详细设计

主菜单设计（图 5-1、5-2）

```
请输入您的用户名:
Administrator
请输入您的密码:
xinglex
登录成功!
您可以选择以下操作:
1、查看当前学生学籍信息
2、修改当前学生学籍信息
3、查看当前班级信息
4、修改当前班级信息
5、检索学生学籍信息
6、修改密码
7、用户管理
0、退出系统
输入您的选择:
```

图 5-1

```
请输入您的用户名:
teacher
请输入您的密码:
teacher
登录成功!
您可以选择以下操作:
1、查看当前学生学籍信
2、查看当前班级信息
3、检索学生学籍信息
4、修改密码
0、退出
请输入您的选择:
```

图 5-2

函数设计及主要功能描述

(1)**public static void log_in(String name, String pwd)**: 用户登录验证。通过传入用户输入的用户名和密码与数据库中的 **users** 表中的用户名和密码一一对照。如果用户名和密码在 **users** 表中，则登录成功。否则登录失败。并且失败次数不得超过三次。超过三次会自动退出系统。同时，如果用户名是 **Administrator**，将标志位 **flag** 设为 **true**。用于后面登录界面的用户的区别。

(2)**public static void student_watch()**: 查看当前学生信息。直接连接数据库。将数据库中的 **student** 表里面的内容全部输出到界面上。

(3)**public static void change_secret()**: 修改密码。用户和管理员均可使用。中间有重复输入新密码并校对两次输入的密码是否相同。

(4)**public static void admin_watch_users()**: 查看当前用户信息，管理员特权。直接连接数据库将数据库中的 **users** 表里面的内容全部输出到界面上。

(7)**public static void admin_create_user()**: 创建新用户，管理员特权。如果有新的用户要求创建用户名，管理员则可以通过此方法创建用户需要的账号。

(6)**public static void admin_delete_user()**: 删除不用的用户名和密码，管理员特权。

(7)**public static void admin_student_insert()**: 插入学生信息，管理员特权。在系统之初，需要向数据库中添加许多学生信息。管理员可以通过此方法完成相应的功能。

(8)**public static void admin_change()**: 修改学生信息，管理员特权。可以对特定学生的各样属性进行修改。

(9)**public static void admin_student_delete()throws Exception**: 删除学生信息，管理员特权。对无用的学生信息进行删除。

(10)**public static void class_watch()throws Exception**: 查看班级信息，公用功能。通过连接数据库，将数据库中 **class** 表输出到界面。

(11)**public static void admin_class_change()**: 修改班级，管理员特权。可以对班级特定的属性进行修改。

(12)**public static void student_search()**: 查找学生，公用功能方法。可以通过学生的各个属性查找特定的学生。然后输出该学生的信息。

(13)**public static void main(String[] args)**: 主方法，整个程序的入口。完成不同方法的正常调用。

6、编码和单元测试

通过详细设计之后，应用程序中所有的方法及其调用方式都已经设计好。接下来就是把详细设计的结果翻译成 **JAVA** 代码。在编写代码的时候，难免会遇到一些问题，大概用了一周的时间编写完成。编写完成扣进行了仔细测试编写出的每一个模块。

各函数测试结果如下图所示

(1) 查看学生信息(图 7-1)

2008302903	付真余	男	4	21	3
2008302904	刘春	男	4	21	7
2008302905	廉作锋	男	4	21	7
2008302907	吴国桢	男	4	21	2
2008302908	李士新	男	4	24	6
2008302909	王小五	男	4	23	5

图 7-1

(2) 插入学生信息(图 7-2)

请输入学生的学号:
2008302915
请输入学生的姓名:
黄文文
请输入学生的性别:
男
请输入学生的班级:
4
请输入学生的年龄:
21
请输入学生的专业:
4
您所要插入的信息为:2008302915 黄文文 男 4 21 4
确认填加信息吗?(y/n):
y
插入成功!

图 7-2

(3) 查看班级信息(图 7-3)

3	计算机3班	硬件	石文兵
4	计算机4班	网络	廖正和

图 7-3

(4)查找特定学生信息(图 7-4)

```
您有如下选择：
    1、按学号查找
    2、按姓名查找
    3、按性别查找
    4、按班级查找
    5、按年龄查找
    6、按专业查找
    0、返回上一级
请输入您的选择：
1
请输入要查找学生的学号：
2008302915
2008302915 黄文文 男 4 21 4
您有如下选择：
    1、按学号查找
    2、按姓名查找
    3、按性别查找
    4、按班级查找
    5、按年龄查找
    6、按专业查找
    0、返回上一级
请输入您的选择：
```

图

(7) 用户操作—删除用户(图 7-5)

```
输入您的选择：
7
您可以有以下操作：
    1、修改管理员密码
    2、查看当前用户
    3、创建新用户
    4、删除用户
    0、退出
请输入您的选择：
4
请输入所要删除用户名：
teacher
删除成功！
```

图 7-5

测试

通过不了解此代码的人通过多次各种各样的测试,发现系统中出现了很多错误。如通过管理员却进入了一般用户的界面、在要求选择的时候出现输入字母后直接退出等等问题。说明此系统还需要用户的进一步使用,也需要我进一步去完善。

维护阶段的关键任务是,通过各和必要的维护活动使系统持久地满足用户的需要。而且是在实际软件设计中花时间最长的一项。但由于此次只是编写课程设计。所以此项未能进行

9、总结

通过此次课程设计,掌握到了数据库系统的开发方法。同时也加深了对刚刚学习的的印象。编写代码的那个星期里面,几乎每天晚上上完自习回来都在编写,总共七个模块,一天大概写一个模块。虽然感觉非常的累,但却也感觉到了一种充实感。有事做总比没事做去打游戏要好的多!整个代码在写的过程中出现了很多问题。有的实在解决不了的问题,就抱着笔记本电脑到下课的时候问教我们 **JAVA**的石文兵老师,老师也非常乐意帮助我解决一些困难。有的时候不懂的问题没必要上网搜,或者自己去看书。因为那样可能用了一上午的时间还不如老师的一句话有用。对这一点我颇有感触。记得书上面只教会了我插入已经给出的内容,但是如果让用户输入要插入的内容,则花了我很长一段时间去搞明白。后来实在找不到解决办法。只能在实验课上问老师了。结果老师花了不到十分钟的时间,不仅给我讲了如何实现插入用户输入的内容。而且还给我讲到了连接数据库的实质以及如何设置数据库的安全性、添加数据库用户等等。而这些之前我只是会把代码抄下来,并没能完全理解。

此次课程设计结束之后,我还意犹未尽,因为从中感觉到了写代码的趣味。于是,我就开始想方设法找点小的应用程序去写。因为这样的学习方法是最快也是最有效的方式。比老师上课讲要好的多。而且大学生也应该培养自己的动手能力。而动手能力只能来自于自己从头到尾的去写一个不大的应用程序。了解整个创作的一个大体思路,在写的过程中还要不断地去请教我们的老师。因为他们对实际应用程序的把握比我们要强的多。

当看到自己写这个 1000 行的 **JAVA**程序能够很顺畅的运行的时候,心里是无比的高兴。也许其它人看到我这个只能在 **DOS** 下运行的程序,感觉非常的不美观;也许他们从网上下的东西比我这好看的多。但是,我想,从中得到的快乐肯定不可同日而语。虽然现在网络很发达,很多东西都能在网上找的到,但是要从中真正学到东西,才能发挥网络的真正作用。像有的同学把源代码下下来后,跑过来问我数据源怎么创建。当然搞的我相当无语。

当然,学无止境,不能因为这么一点小的成就就止步不前。前路很漫长,但我相信只要自己肯用心去学习, **Nothing is impossible!**

王珊,萨师煊. 数据库系统概论[M]. 北京:高等教育出版社,2007.

[2]Java 语言程序设计 北京:清华大学出版社,2005.

[3]SQL Server 2000实验指导(第2版) 北京:清华大学出版社,2008.

11、附源程序

```
import . *;
```

```
public class Student
```

```
{
```

```
    static boolean flag;          //标记是否为管理员
```

```
    static int i = 0;             //限制密码输入次数
```

```
    static String present_name; // 记录当前用户
```

```
    public static void log_in(String name,String pwd) throws Exception//用户登录验证
```

```
    {
```

```
        Connection con = null;
```

```
        Statement stmt = null;
```

```
        ResultSet rs = null;
```

```
        (DBDrive);
```

```
        stmt = ();
```

```
        rs =          * From users where username=          and password=
```

```
        if())
```

```
        {
```

```
            登录成功
```

```
        }
```

```
        else
```

```
        {
```

```
            //设置三秒后关闭程序
```

```

        i++;
        if(i == 3)
        {
            int j;

            for(j=0; j<3; j++)
            {
                (1000);
            }

            if(j == 3)
            {
                (0);
            }

        }
        else
        {
            您的用户名或密码不正确!请重新登录
            您还有      次机会
            Scanner reader = new Scanner;
            请输入您的用户名:
            String username = ();
            请输入您的密码:
            String password = ();
            log_in(username,password);
        }
    }

    {
        flag = true;          //设个管理员的标志位
    }
    else
    {
        flag = false;
    }

}

public static void student_watch()throws Exception    //当前学生信息
{

    Connection con = null;
    Statement stmt = null;

```

```
ResultSet rs = null;
(DBDrive);

stmt = ();

while())
{

}

}

public static void change_secret()throws Exception // _通用 (出现点问题。密码修改不了!)
{
```

```
Connection con = null;
Statement stmt = null;
/*ResultSet rs = null;*/
(DBDrive);

stmt = ();
    您想把密码改为
Scanner reader = new Scanner;
String pwd1 = ();
    请重复新密码
String pwd2 = ();
if(pwd2))
{
    String sql =          users set password =          where username =

    try
    {
        (sql);
    }
    catch(Exception e)
    {
        ;
    }
}
```



```
        密码修改成功！下次登录请用新的密码！
    /*
        rs = (sql1);

        {
            密码修改成功！下次登录请用新的密码！
        }
    */
}
else
{
    两次密码输入不相同。无法完成修改！
}

}

public static void admin_watch_users()throws Exception // 查看当前用户信息_管理员
{

    Connection con = null;
    Statement stmt = null;
    ResultSet rs = null;
    (DBDrive);

    stmt = ();

    while()
    {

    }

}

public static void admin_create_user()throws Exception// 创建用户_管理员
{

    Connection con = null;
    Statement stmt = null;
    ResultSet rs = null;
    (DBDrive);

    stmt = ();
```

```
boolean a = true;
while(a)
{
    Scanner scr = new Scanner();
    请输入用户名:
    String username = scr.next();

    rs = stmt.executeQuery();
    if(rs.next())
    {
        a = false;
        输入有误! 此用户名已经存在!
        break;
    }
    请输入密码:
    Scanner reader = new Scanner();
    String pwd1 = reader.next();
    请重复密码
    String pwd2 = reader.next();
    if(!pwd1.equals(pwd2))
    {
        两次密码输入不相同。无法完成创建!
    }
}

}

public static void admin_delete_user()throws Exception// 删除用户_管理员
{
    Scanner scanner = new Scanner(System.in);
    System.out.println("请输入要删除的用户名:");
    String username = scanner.next();

    rs = stmt.executeQuery();
    if(rs.next())
    {
        System.out.println("该用户不存在!");
    }
    else
    {
        PreparedStatement pstmt = (PreparedStatement) conn.prepareStatement("delete from user where username=?");
        pstmt.setString(1, username);
        int intTemp = pstmt.executeUpdate();
        if(intTemp != 0)
        {
            a = false;
            用户删除成功!
        }
    }
}

}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/058112016132006077>